

AVSによる立体図形の表示

社会学部・情報処理センター

福島万里子・湊 敏

要旨

本報告は、AVS (application visualization system : 3次元可視化ツール) による立体図形表示のためのプログラム開発について述べたものである。AVSはその名が示すようにアプリケーション・プログラムにより得られた数値データをコンピュータ・グラフィックス表示するものである。しかしながら、本学の情報処理センターのコンピュータ・システムにはAVSに適したアプリケーション・プログラムがないので、対称性を持った立体図形表示のためのプログラムをC言語を用いて作成した。対称性を持っている立体という制約があるが、今回作成されたプログラムにより得られる図形をうまく組み合わせることにより、身近にある種々の立体図形をコンピュータ・グラフィックスにより表示できることが示された。

本報告は、AVS (application visualization system : 3次元可視化ツール)¹⁾による立体図形表示のためのプログラム開発について述べたものである。コンピュータはこれまで主として数値・文字データの処理に利用されてきたが、近年のコンピュータのハード・ウェアおよびソフト・ウェアの進歩により画像データや音声データの処理にも幅広く利用されるようになってきた。特に“見る”と言うことは、我々が物事を理解する上で非常に有効な手段である。コンピュータによる図形の視覚化、コンピュータ・グラフィックスは、現実にはないものや我々の目では見えないものを表示することが可能である。このため、コンピュータ・グラフィックスは、都市計画のように今後作られるであろうものの完成予測や今はもうなくなってしまった遺跡の復元、さらに宇宙の様子や分子・原子の世界の様子の描写にも利用されている。また、コンピュータ・グラフィックスは、デザインや芸術の分野にも進出している。このように、コンピュータ・グラフィックスは、コンピュータ利用の1つとして今後ますます発展してゆくものと考えられる。したがって、コンピュータ・グラフィックスのためのソフト・ウェアの開発は意味あるものと考えられる。

AVSは、その名が示す通りグラフィック・プログラムなしでアプリケーション・プログラムにより作成された数値データの可視化を行うことができるソフト・ウェア・システムである。また、入力データとして色々な形態の数値データを取り扱うことができる。可視化にあたっては、イメージ・データ (画像データ) の処理、数値データのグラフ化、立

体図形の処理が行える。立体図形の処理では、着色・材質感・光線の影響を取り込んだレンダリングが可能である。

本報告では、立体図形の処理方法に焦点を絞った。AVSによる立体図形の処理をまとめると図1に示すようになる。まず、AVSにより立体図形を処理するためには立体図形

AVS（3次元可視化ツール）の利用法

現在は、実線の経路で利用

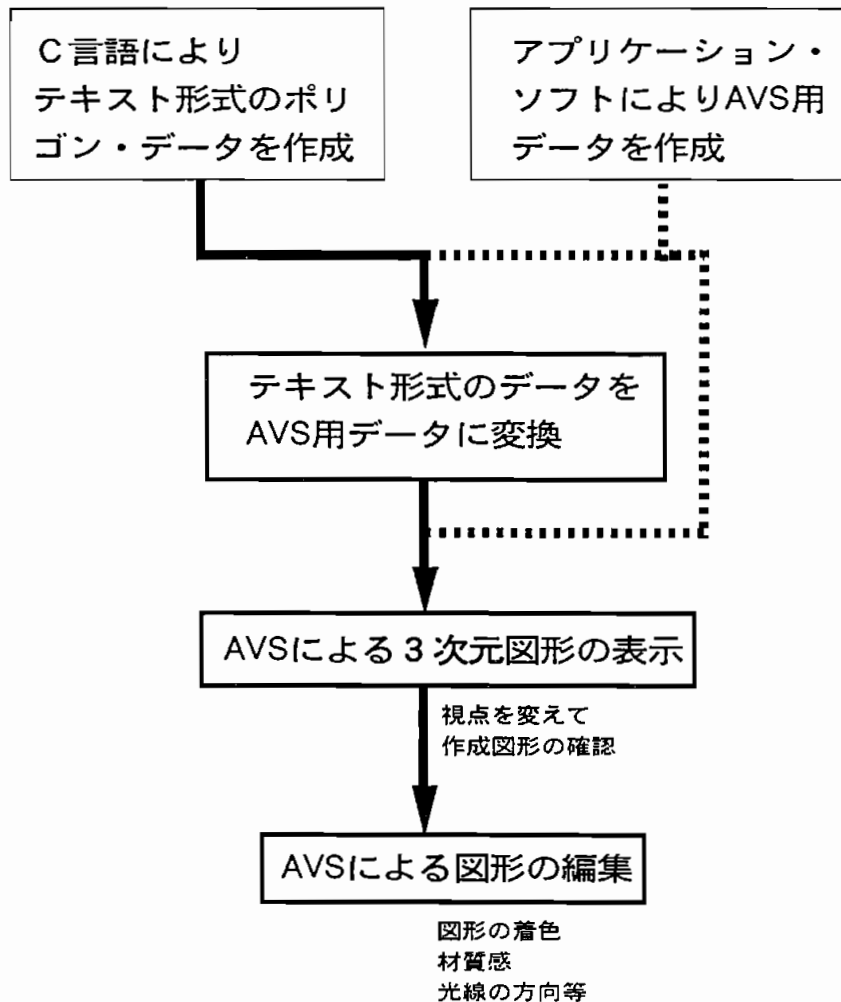


図1. AVSの利用法の流れ図

の空間座標が必要になる。空間座標の作り方には2つの方法があり、第1の方法はアプリケーション・プログラムにより作成する方法であり、第2の方法はAVS利用者が独自で空間座標データを作成する方法である。本学の情報処理センターのコンピュータ・システムにはAVSのためのデータ作成のアプリケーション・プログラムがないので、今回回転の対称性（ C_n ）を持った立体図形の空間座標のためのプログラムをC言語を用いて作成した。

C言語によるプログラムの作成

コンピュータ・グラフィックスによる立体図形の表示では、立体図形は実際にはディスプレイという平面上にしか存在しない。したがってコンピュータ・グラフィックスによる立体表示は、単に立体図形を立体的に見せているだけであると考えがちである。しかしながら、AVSが扱う立体図形は、3次元空間が設定された真の立体図形である。このため、1度立体図形のデータを作成するとその図形はあらゆる方向から見る事が可能になる。

コンピュータ・グラフィックスにより空間における曲面を表示するには、ポリゴン（多角形）を用いる方法がある。ここでは、このポリゴンを用いる方法（ポリゴン法）で立体図形を表示した。このポリゴン法の例を図2に示した。

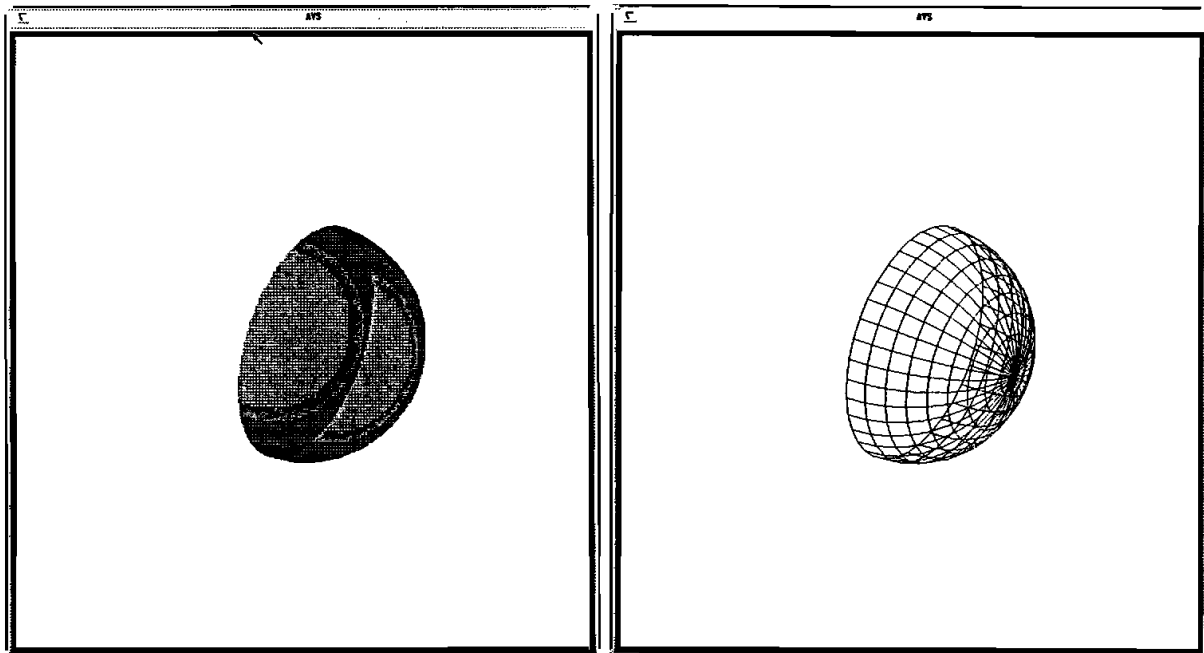


図2. AVSによる半球面の表示例

左図は、半球面を示す。右図は半球面を構成するポリゴンを示す。

例えば、図2の左は半球面を表している。この半球面は、図2の右の図に示したように多くの三角形と四角形から構成されている。これらの全ての三角形と四角形は曲面でなく平面である。即ち、ポリゴン法による空間の曲面は、多角形（ここでは三角形と四角形）の平面で作られている。このため、曲面らしく見せるためには多くの多角形を空間に作る必要がある。もちろん、人間がこれらの多角形のデータを作ることは可能であるが、実際に

多くのポリゴンを作成するためには多大の時間を要する。このため、このポリゴンのデータを作るプログラムの作成を行った。プログラムの作成にあたっては本学情報処理センターのCONVEX C-3420コンピュータを用い、プログラム言語としてはC言語を用いた。また、立体図形の表示は、CONVEX C-3420コンピュータに導入されている3次元可視化ツール、AVSを用いて行った。

AVSは真の立体図形を扱うため3次元空間の設定、即ち、空間座標が必要になる。図3にAVSにおけるディスプレイ上の空間座標を示した。座標の原点はディスプレイの

AVSの座標系

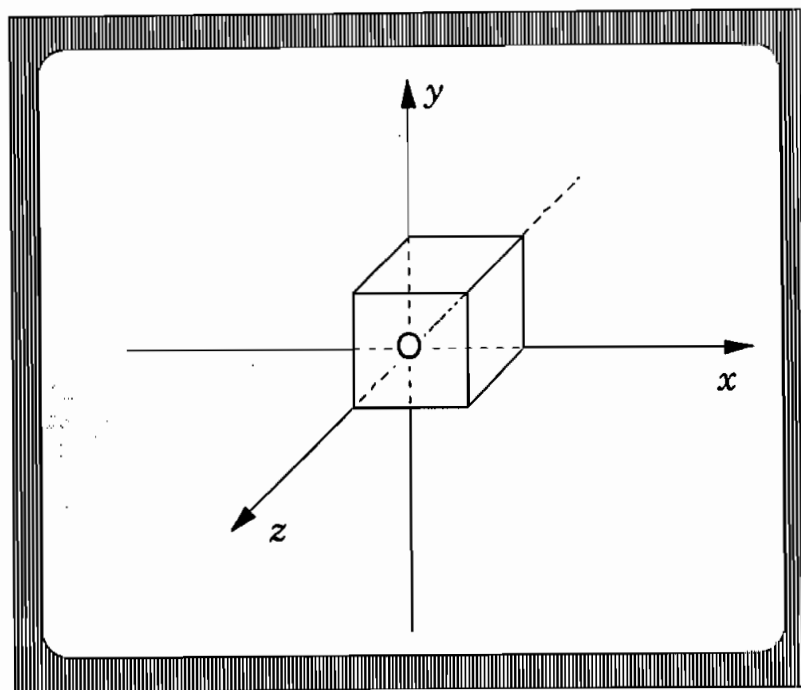


図3. AVSにおけるディスプレイ上の座標系

中央にあり、ディスプレイ平面は空間座標のx y平面に対応している。ディスプレイの左右端のx座標はほぼ±5.0であり、上下端のy座標はほぼ±5.0である。また、ディスプレイの手前はz座標で正の方向であり、ディスプレイの裏側はzの座標で負の方向である。例えば、図3に示された1つの頂点を原点においた1辺が2.0の立体の空間座標は(0.0, 0.0, 0.0), (2.0, 0.0, 0.0), (2.0, 2.0, 0.0), (0.0, 2.0, 0.0), (0.0, 0.0, 2.0), (2.0, 0.0, 2.0), (2.0, 2.0, 2.0), (0.0, 2.0, 2.0)となる。

AVSでは、アスキー形式で作成されたテキストデータをAVS用のグラフィックデータに変換するフィルター・プログラム (polyg_to_geom) を持っている。

smooth				常に必要
4				多角形の頂点の数
0.0	0.0	0.0		頂点の X, Y, Z 座標
0.0	2.0	0.0		頂点の X, Y, Z 座標
2.0	2.0	0.0		頂点の X, Y, Z 座標
2.0	0.0	0.0		頂点の X, Y, Z 座標

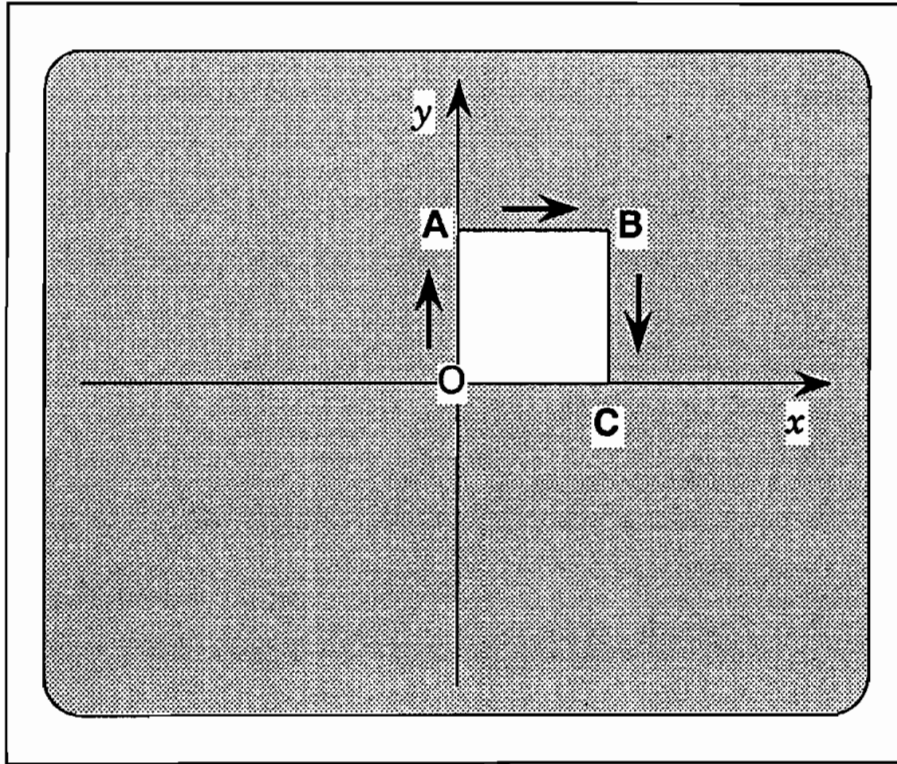


図4. フィルター・プログラム (polyg_to_geom)
の入力データの例とその座標系

図4に1辺が2.0の正方形を描く場合のこのフィルター・プログラム、**polyg_to_geom**、のデータ形式と座標系を示した。ここで、注意することはポリゴンの面に“表”と“裏”があることである。AVSでは、立体図形に光線をあてることができ、その方向も自由に変えることができる。光線をあてられた立体図形は、光線のあたっている部分（光る）と影の部分（光らない）とに分けられる。ポリゴンの表とは、光線をあてたとき光る面のことである。したがって、ポリゴンの裏側に光をあてても光ることはない。さて、入力データの最初はポリゴンの結合の形式であり、“smooth”と“facet”が選べる。smoothとすればポリゴン同士が滑らかにつながり、曲面を表すときに用いる。facetとすればポリゴン同士が角張ってつながる。例では、smoothとしてある。次に必要なデータは多角形の頂点の数である。ここでは、正方形を例にとってあるので4である。次に4つの頂点の

空間座標が示してある。ここで、4つの頂点の空間座標の入りの順がポリゴンの面の表と裏を決める。頂点の座標を順に結んだとき、これらの点の順番が時計まわりになっていると見ている面がポリゴンの裏の面になる（即ち、右ねじが進む方向が表になる）。図4に示した正方形では紙面側がポリゴンの裏になり、紙面の裏側がポリゴンの表になる。ここでは1つの正方形平面の例を示したが、実際に立体を作るときは、立体を構成するポリゴンの頂点の数と頂点の座標をこのデータの後ろに必要とするポリゴンの数だけ続けて入力してゆけばよい。

本報告では、回転の対称軸を持った立体のアスキー形式のポリゴン・データを作成するプログラムは、`rot`、と切り立が楕円になる立体のアスキー形式のポリゴン・データを作成するプログラム、`ellipes`、の2つを紹介する。これらのソース・プログラムは、付録に示した。もちろん、これらの2つのプログラムで作られたデータは、上に述べたフィルター・プログラム、`polyg_to_geom`、によりAVS用のデータに変換する必要がある。`rot`では、立体を幾つかの水平面で切った時の切り口の面の集まりで表す。必要な入力データは立体を切る水平面の数とその切り口面の多角形の頂点の数、および切り口の面のy座標と大きさ（切り口の多角形の外接円の半径）である。図5に入力データの例を示した。

右の立体の入力データ

2		面の数
4		多角形の頂点の数
0.0	3.0	y座標と外接円の半径
2.5	2.0	y座標と外接円の半径

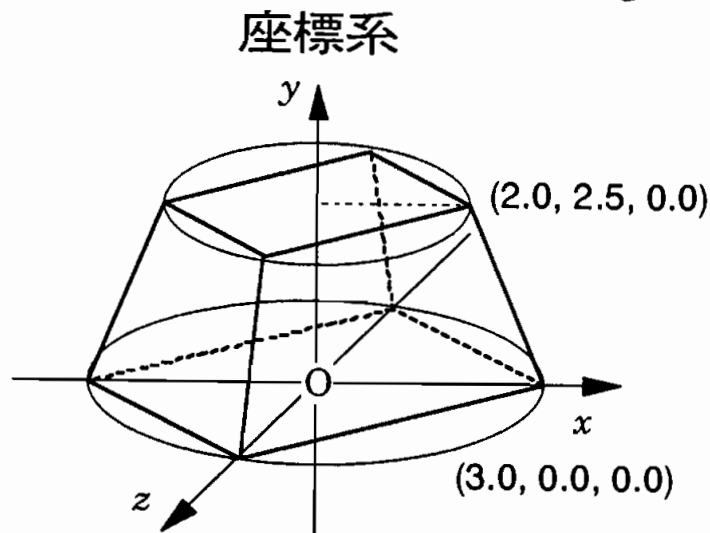
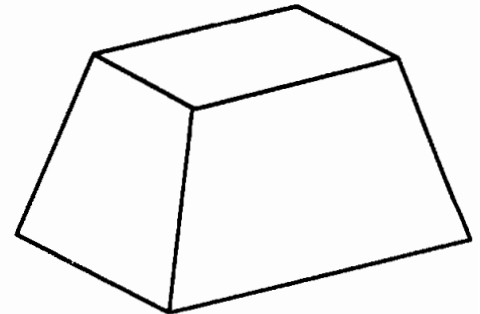


図5. `rot`の入力データの例とその座標系

ellipseも立体を幾つかの水平面で切った時の切り口の面の集まりで表す。必要な入力データは立体を切る水平面の数とその切り口の楕円面を作る頂点数（例では8）、および切り口の面の y 座標と楕円の長軸と短軸の長さである。図6に入力データの例を示した。

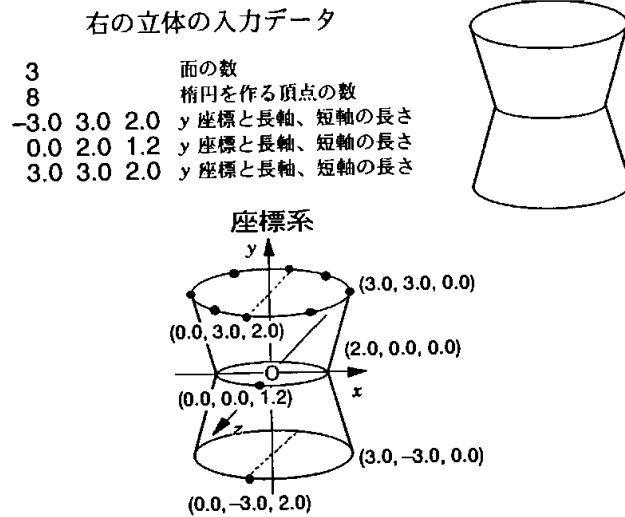


図6. ellipseの入力データの例とその座標系

立体の表示例

図7にrotを用いて回転の対称軸を持った6つの図形から構成された石塔の例²⁾を示した。白い線は光の照射方向を示している。光は右上方からあたっている。一番下の立体は水平に切ったときの切り口が正方形のものである。2つ上の立体は丸く見えているが、この切り口は正24角形である。この立体の切り口は多角形であるが、見かけ上充分丸く見えている。

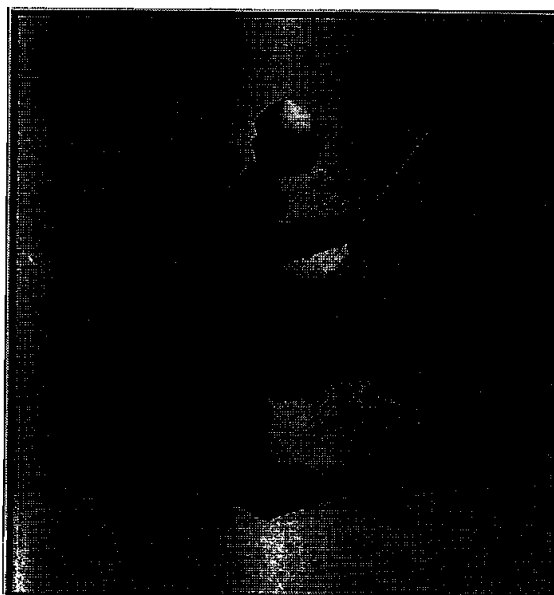


図7. AVSによる石塔の表示

図8にワイングラスを示した。光は右上からあっている。ここで注意することは、ワイングラスのワインの入る部分は2つの面から構成されていることである。先に示したように面には表と裏があり、光をあてた場合、表側は光るが裏側は光らない。このため、ワインの入る部分を1つの面で表すと光を照射したとき、光るべき内側が光らなくなる。このため、このワイングラスのワインの入る部分は2つの面で作られている。

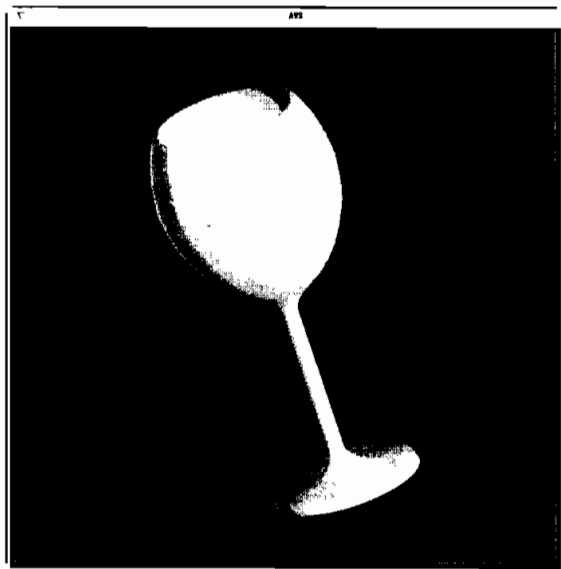


図8．AVSによるワイングラスの表示

図9は壺を示してある。光は右上方からあっている。この壺ではワイングラスのワインの入る部分と同様に全ての面は2つの面から構成されている。アクセントとしてギザギザの模様を入れてある。

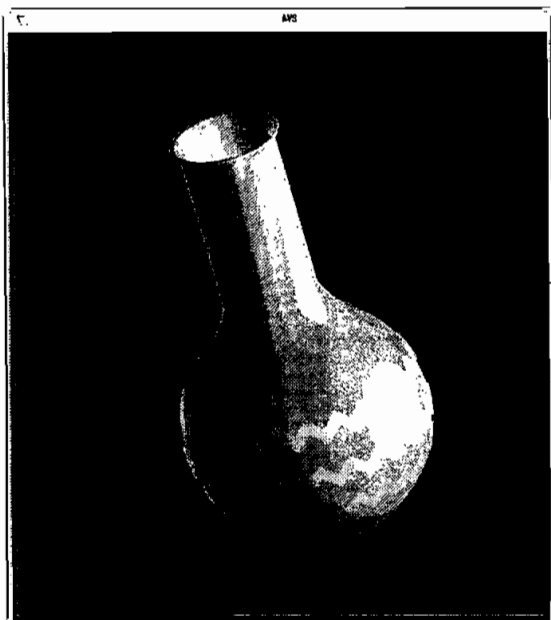


図9．AVSによる壺の表示

図10はrotとellipseを用いて作成した砂時計を示してある。光はこれまでと同様に右上方からあたっている。砂時計の上下の部分はellipseを用いて作成してある。残りの部分はrotを用いて作成してある。ただ横の2つの軸は円柱、即ちrotで作られる円柱をずらせただけである。この砂時計の上下の部分と2つの軸は中心の砂の入る部分と比べると光線の向きに対して濃淡にあまり差がでていない。これは上下の部分と軸の部分（木製と考えた）は中心部分（ガラスと考えた）と材質感を変えてあるためである。

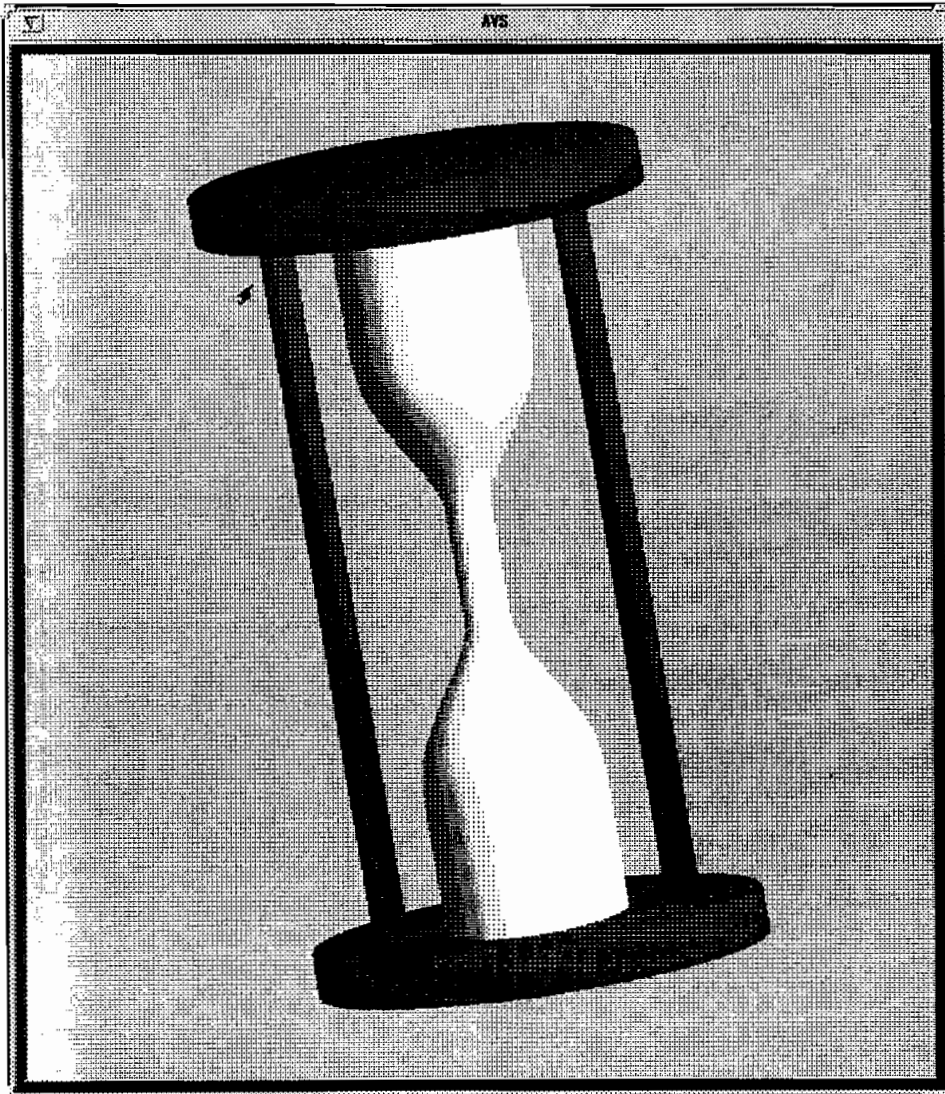


図10. AVSによる砂時計の表示

以上、今回作成したプログラムを用いて作成した立体図形の4つの例を示した。今回作成した立体図形は、3次元の図形で着色されたものである。この報告では、色々な方向から見るといった動きや光線の照射方向の変更等を紹介できないのは残念である。また、着色された図形は白黒で印刷されているため、色合いが表現できないのが残念である。

おわりに

本報告で紹介した2つのプログラムは、対称性を持った立体作成のためのデータしか作ることにはできない簡単なものである。しかしながら、これらの2つのプログラムで作成された立体をうまく組み合わせることにより、我々の身近にあるものをコンピュータ・グラフィックスにより表示できることが分かった。本報告では示さなかった曲面を持っていない立体図形のデータは簡単に作ることができるので、ここで紹介したプログラムで作られた図形と組み合わせることによりさらに多くの立体図形、例えば建造物等をコンピュータ・グラフィックスにより表示できると考えられる。

コンピュータ・グラフィックスによる3次元表示の大きな利点は、立体図形を見たい方向から見るができることと立体に色々な色の光があたったときどの様に見えるかが分かることである。本学のCONVEX C-3420コンピュータは高速であるためこのような処理を瞬時に行ってくれる。CONVEX C-3420コンピュータ上でAVSを利用することは立体図形の理解に非常に有利であると考えられる。以上のことから、AVSを有効に利用すれば、AVSは研究・教育面で非常に有効な道具になると考えられる。

謝 辞

本報告を作成するにあたって、CONVEX C-3420コンピュータの計算機使用時間を与えてくださった奈良大学情報処理センターに感謝する。

脚 注

- 1) AVSは、Advanced Visual System社の登録商標です。
- 2) 本学文学部文化財学科神田周枝の作品である。

Oct 19 15:46 1993 kaiten.c Page 1

【付録1】

回転体のデータ作成のためのプログラム

```

#include <stdio.h>
#include <math.h>

void main(int argc,char *argv[])

    /*          -----          */
    /*      回転体のデータ作成      */
    /*          -----          */
{
    double px[101],py[101],pz[101];
    double qx[101],qy[101],qz[101];
    double theta,pi,inc,rdown,rup,hdown,hup;
    double r[50],hight[50];
    int i,j,m,n;
    char out_file[80];
    FILE *in_fp, *out_fp;

    if (argc ==1) {
        puts("\n >>>> Error <<<<<");
        puts("\n Usage * %geom3d filename(grid data)");
        exit(-1);
    }
    if((in_fp = fopen(argv[1],"r")) ==NULL) {
        puts("\n can't open the file");
    }

    strcpy(out_file,argv[1]);
    strcat(out_file, ".data");

    out_fp = fopen(out_file,"w");

    pi=atan(1.0)*4.0;

    fprintf(out_fp,"smooth\n");

    /*  切り口の面の数          */
    fscanf(in_fp,"%d",&n);

    /*  切り口の頂点の数          */
    fscanf(in_fp,"%d",&m);

    /*  切り口の y 座標と切り口の大きさ          */
    for (i=1;i<=m+1;i++)
        fscanf(in_fp,"%lf %lf",&hight[i],&r[i]);

    inc=360.0/n;
    inc=inc*pi/180.0;

```

Oct 19 15:46 1993 kaiten.c Page 2

```

for (i=1;i<=100;i++)
  ( px[i]=0.0;
    py[i]=0.0;
    pz[i]=0.0;
    qx[i]=0.0;
    qy[i]=0.0;
    qz[i]=0.0;
  )
for (j=1;j<=m;j++) (
  rdown=r[j];
  rup=r[j+1];
  hdown=hight[j];
  hup=hight[j+1];

theta=0.0;
for (i=1;i<=n;i++)
  ( px[i]=cos(theta)*rdown;
    py[i]=hdown;
    pz[i]=sin(theta)*rdown;
    qx[i]=cos(theta)*rup;
    qy[i]=hup;
    qz[i]=sin(theta)*rup;
    theta=theta+inc;
  )

  px[n+1]=px[1];
  py[n+1]=py[1];
  pz[n+1]=pz[1];
  qx[n+1]=qx[1];
  qy[n+1]=qy[1];
  qz[n+1]=qz[1];

for (i=1;i<=n;i++)
  ( fprintf(out_fp," 4\n");
    fprintf(out_fp," %10.7f %10.7f %10.7f\n",px[i],py[i],pz[i]);
    fprintf(out_fp," %10.7f %10.7f %10.7f\n",qx[i],qy[i],qz[i]);
    fprintf(out_fp," %10.7f %10.7f %10.7f\n",qx[i+1],qy[i+1],qz[i+1]);
    fprintf(out_fp," %10.7f %10.7f %10.7f\n",px[i+1],py[i+1],pz[i+1]);
  )
}

theta=0.0;
for (i=1;i<=n;i++)
  ( px[i]=cos(theta)*r[1];
    py[i]=hight[1];
    pz[i]=sin(theta)*r[1];
    qx[i]=cos(theta)*r[m+1];
    qy[i]=hight[m+1];
    qz[i]=sin(theta)*r[m+1];
    theta=theta+inc;
  )

  px[n+1]=px[1];
  py[n+1]=py[1];
  pz[n+1]=pz[1];
  qx[n+1]=qx[1];

```

Oct 19 15:46 1993 kaiten.c Page 3

```
    qy[n+1]=qy[1];
    qz[n+1]=qz[1];

    fprintf(out_fp," %3d\n",n);
    for (i=1;i<=n;i++)
        fprintf(out_fp," %10.7f %10.7f %10.7f\n",px[i],py[i],pz[i]);

    fprintf(out_fp," %3d\n",n);
    for (i=n;i>=1;i--)
        fprintf(out_fp," %10.7f %10.7f %10.7f\n",qx[i],qy[i],qz[i]);

    fclose(in_fp);
    fclose(out_fp);
}
```

[付録 2]

楕円体のデータ作成のためのプログラム

```
#include <stdio.h>
#include <math.h>

void main(int argc, char *argv[])

    /*          -----          */
    /*          楕円体のデータ作成          */
    /*          -----          */
{
    double px[101], py[101], pz[101];
    double qx[101], qy[101], qz[101];
    double theta, pi, inc, r1down, r2down, r1up, r2up, hdown, hup;
    double r1[50], r2[50], hight[50];
    int i, j, m, n;
    char out_file[80];
    FILE *in_fp, *out_fp;

    if (argc == 1) {
        puts("\n >>>> Error <<<<<");
        puts("\n Usage * %geom3d filename(grid data)");
        exit(-1);
    }
    if((in_fp = fopen(argv[1], "r")) == NULL) {
        puts("\n can't open the file");
    }

    strcpy(out_file, argv[1]);
    strcat(out_file, ".data");

    out_fp = fopen(out_file, "w");

    pi = atan(1.0) * 4.0;

    fprintf(out_fp, "smooth\n");

    /*          切り口の面の数          */
    /*          */

    fscanf(in_fp, "%d", &n);

    /*          楕円を作る頂点の数          */
    /*          */

    fscanf(in_fp, "%d", &m);

    /*          楕円の y 座標と長軸および短軸の長さ          */
    /*          */

    for (i=1; i<=m+1; i++) fscanf(in_fp, "%lf %lf %lf", &hight[i], &r1[i], &r2[i]);

    inc = 360.0/n;
    inc = inc * pi / 180.0;

    for (i=1; i<=100; i++)
```

Oct 19 15:43 1993 daentai.c Page 2

```

    { px[i]=0.0;
      py[i]=0.0;
      pz[i]=0.0;
      qx[i]=0.0;
      qy[i]=0.0;
      qz[i]=0.0;
    }
  for (j=1;j<=m;j++) {
    r1down=r1[j];
    r1up=r1[j+1];
    r2down=r2[j];
    r2up=r2[j];
    hdown=hight[j];
    hup=hight[j+1];

    theta=0.0;
    for (i=1;i<=n;i++)
      { px[i]=cos(theta)*r1down;
        py[i]=hdown;
        pz[i]=sin(theta)*r2down;
        qx[i]=cos(theta)*r1up;
        qy[i]=hup;
        qz[i]=sin(theta)*r2up;
        theta=theta+inc;
      }

    px[n+1]=px[1];
    py[n+1]=py[1];
    pz[n+1]=pz[1];
    qx[n+1]=qx[1];
    qy[n+1]=qy[1];
    qz[n+1]=qz[1];

    for (i=1;i<=n;i++)
      { fprintf(out_fp," 4\n");
        fprintf(out_fp," %10.7f %10.7f %10.7f\n",px[i],py[i],pz[i]);
        fprintf(out_fp," %10.7f %10.7f %10.7f\n",qx[i],qy[i],qz[i]);
        fprintf(out_fp," %10.7f %10.7f %10.7f\n",qx[i+1],qy[i+1],qz[i+1]);
        fprintf(out_fp," %10.7f %10.7f %10.7f\n",px[i+1],py[i+1],pz[i+1]);
      }
  }

  theta=0.0;
  for (i=1;i<=n;i++)
    { px[i]=cos(theta)*r1[1];
      py[i]=hight[1];
      pz[i]=sin(theta)*r2[1];
      qx[i]=cos(theta)*r1[m+1];
      qy[i]=hight[m+1];
      qz[i]=sin(theta)*r2[m+1];
      theta=theta+inc;
    }

    px[n+1]=px[1];
    py[n+1]=py[1];
    pz[n+1]=pz[1];

```

Oct 19 15:43 1993 daentai.c Page 3

```
qx[n+1]=qx[1];
qy[n+1]=qy[1];
qz[n+1]=qz[1];

fprintf(out_fp," %3d\n",n);
for (i=1;i<=n;i++)
  fprintf(out_fp," %10.7f %10.7f %10.7f\n",px[i],py[i],pz[i]);

fprintf(out_fp," %3d\n",n);
for (i=n;i>=1;i--)
  fprintf(out_fp," %10.7f %10.7f %10.7f\n",qx[i],qy[i],qz[i]);

fclose(in_fp);
fclose(out_fp);
}
```